Протокол до Домашньої роботи N°2



Андрій Полішко

Distributed map

1. Сторення та запис до розподіленої мапи

```
import hazelcast

import hazelcast.

client = hazelcast.HazelcastClient(
 cluster_name="hello-world",
)

# Create a Distributed Map in the cluster

map = client.get_map("my-distributed-map").blocking()

values = ['BMW', 'Mercedes', 'Audi', 'Toyota', 'Honda', 'Ford', 'Nissan', 'Tesla', 'Hyundai', 'Kia']

for i in range(1000):
 map.put(str(i), values[i % len(values)])
```

2. Розподіл даних у Managment Center



- 3. Зміни розподілу за різних обставин
 - а. При відключенні однієї ноди(третьої)



b. При відключенні двох (другої та третьої)

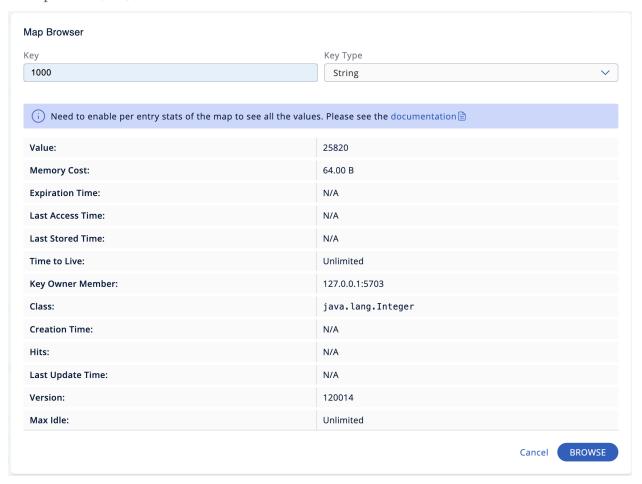


Як бачимо на останньому скріншоті, втрати даних не відбулось. Це

досягається за допомогою реплікації.

Distributed map with and without locks

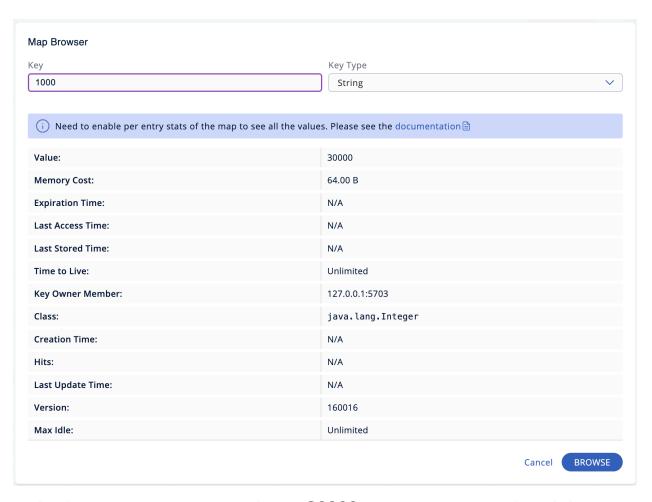
1. Значення при інкременті 10000 разів трьома клієнтами без механізмів синхронізації одночасно



Як бачимо, значення не дорівнює 30000, через race condition.

Код використаний для цього прикладу знаходиться у теці ./raceConditionTest/notLockedExamples

2. Аналогічний тест з використанням блокування нашої змінної (Pessimistic locking)



Як і очікувалося, значення дорівнює 30000, тому що при кожній зміні значення, воно блокувалось поточним клієнтом (це також відобразилось на швидкодії, такий код був відчутно повільніший)

Код використаний для цього прикладу знаходиться у теці ./raceConditionTest/pessimisticLockedExamples

3. При оптимістичному блокуванні результат, очікувано, аналогічний Код використаний для цього прикладу знаходиться у теці ./raceConditionTest/optimisticLockedExamples

Bounded queue

Результати декількох запусків одного продюсера та двох конс'юмерів

Так як ми поставили побмеження на 10 елсентів до нашої черги, то відповідно запускати продюсера потрібно декілька разів, томущо кожен консюмер очікує по 100 елементів.

Конс'сюмери забирають елемети з черги по черзі, якщо запущені паралельно.

Якщо консюмери не запущені, а черга переповнилась, то просто не додаються нові елементи.